



Kitson, J., Williamson, S., Harper, P., McMahon, C., Rosenberg, G., Tierney, M., & Bell, K. (2017). A photovoltaic panel modelling method for flexible implementation in Matlab/Simulink using datasheet quantities. In *2017 26th IEEE International Symposium on Industrial Electronics (ISIE 2017)* (pp. 946-951). (Proceedings of the International Symposium on Industrial Electronics). Institute of Electrical and Electronics Engineers (IEEE).
<https://doi.org/10.1109/ISIE.2017.8001373>

Peer reviewed version

Link to published version (if available):
[10.1109/ISIE.2017.8001373](https://doi.org/10.1109/ISIE.2017.8001373)

[Link to publication record in Explore Bristol Research](#)
PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via IEEE at <http://ieeexplore.ieee.org/document/8001373/>. Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

A photovoltaic panel modelling method for flexible implementation in Matlab/Simulink using datasheet quantities

J. Kitson^{1*}, S. J. Williamson¹, P. Harper¹, C. M. McMahon², G. Rosenberg¹, M. Tierney¹, K. Bell¹

¹University of Bristol, Bristol, UK

²DTU Technical University of Denmark, Kongens Lyngby, Denmark

*jk6862@bristol.ac.uk

Abstract—This paper presents a detailed method for creating an embedded Matlab model in Simulink for any solar photovoltaic panel starting with its datasheet values. It links extrinsic functions to the Simulink embedded model to provide fast and simple iterative solving of non-linear equations. It also provides a method sufficiently flexible to produce a model output based on panel current or voltage such that it can be cascaded with different Simulink elements.

Keywords—Solar photovoltaic, embedded Matlab modelling, Simulink, Newton Raphson, simultaneous non-linear equations, Jacobian matrices, extrinsic and anonymous Matlab functions

NOMENCLATURE

Symbol	Description
A	Diode quality (ideality) factor
I_o	Dark saturation current in Standard Test Conditions (STC)
I_{sc}	Short-circuit current in STC
I_{mpp}	Current at the Maximum Power Point (MPP) in STC
k	Boltzman constant (J/K)
K_i	Temperature coefficient of the short-circuit current
K_v	Temperature coefficient of the open-circuit voltage
N_s	Number of cells per module
P_{mpp}	Power at the MPP in STC
q	Charge on an electron
R_s	Panel series resistance
R_{sh}	Shunt resistance
T_{stc}	Temperature at STC (°K)
V_{mpp}	Voltage at the MPP in STC
V_{oc}	Open-circuit voltage in STC
V_t	Thermal voltage on photovoltaic cell

I. INTRODUCTION

Solar photovoltaic installations have increased in number dramatically over the past 10 years, with the cost of panels reducing by 80% since 2010, making large solar installations a viable commercial alternative to fossil fuel power plants [1]. New photovoltaic materials continue to improve panel efficiency considerably [2]. This has been coupled with increases in grid-tie inverter efficiency [3] and control improvements to reduce harmonics whilst maintaining fast response [4], making PV increasingly attractive for use in DC or hybrid DC/AC grids.

Where PV panels are to be incorporated in a DC or hybrid DC/AC grid or micro-grid system in development, it is important that a robust but flexible means of simulating modelling a PV panel in software such as Matlab/Simulink is available for system specification, panel sizing and maximum power point tracking design.

Simulink provides in-built models of PV panels with drop down options for different panels; however it may not provide a match for a desired model; built-in model operation tends to be opaque and based on look-up tables. When coupled with other system elements, they can form algebraic loops which are challenging to resolve. They are limited in terms of how they must be connected with other Simulink model parts; this can therefore make them difficult to cascade with other elements of simulated Microgrid systems.

In order to provide simpler, more robust and more flexible solar models, an alternative approach to Simulink in-built models can be used which characterizes the current-voltage behaviour of PV panels from first principles but which can be dropped into a variety of Simulink systems.

This paper provides a step by step process for modeling of any PV panel in embedded Matlab/Simulink from a datasheet with the option of having panel current or voltage as a model output. This comprises an illustration of (1) how equivalent circuit parameters are derived from solving a series simultaneous of non-linear equations for the panel at standard conditions (STC) through Matlab functions (2) how an embedded Matlab function within Simulink can be created to represent the PV panel with the choice of panel output current being a model input or output (similarly with panel voltage); and (3) how the Simulink embedded Matlab function can call extrinsic iterative solvers (e.g. Newton-Raphson) and return values to the embedded function for output into Simulink model. Finally (4), the model outputs are compared with plotted $v-i$ characteristics of panels found in datasheets for a range of solar panel models to verify their agreement with manufacturers' values over a range of insolation and temperature values.

II. PV EQUIVALENT CIRCUIT MODEL DEVELOPMENT

A. Choice of Equivalent Circuit Model for a PV Panel

There have been a number of modelling options starting from equivalent circuit equations which have been researched for PV cells, trading off accuracy with complexity [9], [10], [11], [12], [13] and [14]. The simplest of these models is based on a single current source and diode, extending to a double exponential model with equivalent series and shunt resistances. This work will use one such model in the range; the option selected is a standard single exponential model with a series and shunt resistance, [8], [11] and [12] shown in Fig 1. The same modelling and iterative solving approaches set out in this paper are equally applicable to other more complex equivalent circuit models, with appropriate alterations in underlying circuit equations.

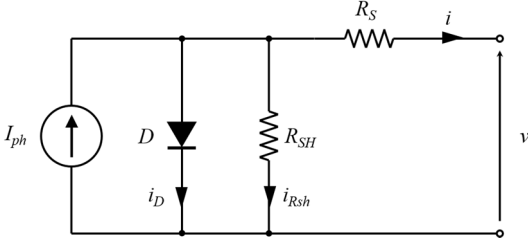


Fig 1. Equivalent Circuit Model based on a single diode with shunt and series resistor [8], [11] and [12]

B. Equivalent Circuit Parameters for Shunt and Series Resistances and Diode Ideality from Datasheet Values

The equivalent circuit model equations can be found by applying Kirchhoff's voltage and current laws to the equivalent circuit model in Fig 1; and these have been set out in detail previously in the literature [11], [13]. The equivalent circuit equations used in this work are based primarily on the derivation set out in [11]. These V - I relationships are shown in (1) and (2).

$$i = I_{ph} - I_0 \left(e^{\frac{v+iR_s}{n_s V_t}} - 1 \right) - \frac{v+iR_s}{R_{sh}} \quad (1)$$

$$v = -R_{sh} \left[i - I_{ph} + I_0 \left(e^{\frac{v+iR_s}{n_s V_t}} - 1 \right) \right] - iR_s \quad (2)$$

The equivalent circuit parameters R_{sh} , R_s , A can be derived by looking at the three points of interest in the panel circuit V - I curve. At open circuit where no current flows, (3) applies [11].

$$I_{oc} = 0 = I_{ph} - I_0 e^{\frac{V_{oc}}{n_s V_t}} - \frac{V_{oc}}{R_{sh}} \quad (3)$$

At short circuit where there is no voltage across the panel, therefore (4) is applicable.

$$I_{sc} = I_{ph} - I_0 e^{\frac{R_s I_{sc}}{n_s V_t}} - \frac{I_{sc} R_s}{R_{sh}} \quad (4)$$

Finally, at maximum power point (5) can be derived [11]

$$I_{mpp} = I_{ph} - I_0 e^{\frac{V_{mpp} + R_s I_{mpp}}{n_s V_t}} - \frac{I_{mpp} R_s}{R_{sh}} \quad (5)$$

Equation (6) shows the relationship between thermal voltage at standard conditions $V_{t_{stc}}$ and the diode ideality factor, A .

$$V_{t_{stc}} = \frac{AkT_{stc}}{q} \quad (6)$$

Equations (3)-(6) are used in conjunction with the measurements typically provided solar PV panel datasheets, (TABLE I.). Further equation manipulation via the process outlined in [11] yields three non-linear equations (equations (18), (19 and (22) in [11]) which contain expressions for the known datasheet parameters in TABLE I. but in which three unknowns remain: R_s , R_{sh} and A .

TABLE I. DATASHEET VALUES FOR A SELECTION OF COMMERCIALY AVAILABLE PV PANELS AT STANDARD CONDITIONS (STC)

Datasheet Params	Make and Model of Solar Panel				
	SLP020	MSX120 (24V)	MSX60	MSX64	KC200GT
Isc (A)	1.31	3.87	3.8	4.0	8.21
Voc (V)	21.6	42.1	21.1	21.3	32.9
Impp (A)	1.2	3.52	3.5	3.66	7.66
Vmpp (V)	18.6	33.7	17.1	17.5	26.3
Ns	36	72	36	36	54
ki (%/°C)	0.065	0.065	0.065	0.065	3.18 mA/°C
kv (mV/°C)	-65	-160	-80	-80	-123

C. Solving a set of non-linear simultaneous equations for equivalent circuit parameters using Jacobian Matrices

In order to solve the set of non-linear equations arising to extract the equivalent PV circuit model parameters, R_s , R_{sh} and A , a number of methods have been proposed. These have been, for example, via a pattern search optimization algorithm [12], nested loops to solve for several circuit variables using an iterative Newton-Raphson or bisection method [11], or by using Lambert w-functions [13]. This work derives circuit parameters by solving the non-linear equations simultaneously with a Newton Raphson method using a Jacobian matrix [15].

A system of n simultaneous equations with n unknowns can be solved using a variation on the iterative Newton-Raphson method. In this case the value of three unknowns to be solved for R_{sh} , R_s , A should be such that they can satisfy (7).

$$\begin{bmatrix} f_1(R_{sh}, R_s, A) \\ f_2(R_{sh}, R_s, A) \\ f_3(R_{sh}, R_s, A) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (7)$$

Based on the equation derivation in [11] at the three points of interest in the V - I output characteristic of the solar panel, the three nonlinear equations (15)-(17) to be solved iteratively in this work, have been rearranged such that they meet the form set out in (7). This iterative solution is reached by using a Newton-Raphson progression with a Jacobian matrix as in (8).

$$x_{n+1} = x_n - J(x_n)^{-1}f(x_n) \quad (8)$$

The Jacobian matrix is defined as a matrix of partial derivatives of the three equations f_1, f_2 and f_3 as in (9).

$$J = \begin{bmatrix} \frac{\delta f_1}{\delta R_{sh}} & \frac{\delta f_1}{\delta R_s} & \frac{\delta f_1}{\delta A} \\ \frac{\delta f_2}{\delta R_{sh}} & \frac{\delta f_2}{\delta R_s} & \frac{\delta f_2}{\delta A} \\ \frac{\delta f_3}{\delta R_{sh}} & \frac{\delta f_3}{\delta R_s} & \frac{\delta f_3}{\delta A} \end{bmatrix} \quad (9)$$

The progression in (8) must be initialized with estimates of the three parameters to be solved, as expressed in (10).

$$x_0 = [R_{sh0} \quad R_{s0} \quad A_0]^T \quad (10)$$

Seed values for these were set where $R_{sh0} \approx 1000$, $R_{s0} \approx 0.3$ and $A_0 \approx 1.1$ for the Newton-Raphson iterative process to converge. The solution can either be implemented by writing anonymous functions for $f(x_n)$ and $J(x_n)^{-1}$ in one Matlab file, or by writing separate functions for each of $f(x_n)$ and $J(x_n)^{-1}$ and including those functions as parameters in another function (called 'newtonPVparam'), Fig 2:

```
function [x,iter] =
newtonmPVparam(x0, PVparam, f, J)
N = 100; % define max. number of iterations
epsilon = 1e-7; % define tolerance
xx = x0; % load initial guess
while (N>0) JJ = feval(J,xx,PVparam);
xn=xx - JJ\feval(f,xx,PVparam);
if abs(feval(f,xn,PVparam))<epsilon
x=xn; iter = 100-N;return; end;
N = N - 1;xx = xn; end;
```

Fig 2. Matlab .m file implementation of (8)

This function accepts the initial values of x (as expressed in (10), together with the datasheet parameters "PVparam" and calls two parameterised functions 'nonlinearfunctions' which relates to $f(x_n)$ and 'jacob' which refers to $J(x_n)^{-1}$. It returns one vector with the solutions for the three circuit values R_{sh} , R_s and A ; these are shown in TABLE II. V_{tstc} (thermal voltage at standard conditions) is directly proportional to A (see (6)) and has therefore also been included in the same results table.

TABLE II. CALCULATED EQUIVALENT CIRCUIT PARAMETERS AT STC USING VALUES 1-5 FROM TABLE I.

Calculate d Circuit Params	Make and Model of Solar Panel				
	SLP020	MSX120	MSX60	MSX64	KC200 GT
Rsh	449.15	940.9	1975.8	1044.5	15153
Rs	0.0002	0.4539	0.311	0.196	0.3279
A	1.1132	1.4707	1.2265	1.3271	0.9821
V _{tstc}	0.0286	0.0378	0.0315	0.0341	0.0253

D. Dependence of Circuit Equations on Insolation and Temperature

Once ascertained, the parameters R_s , R_{sh} and A are assumed to be fixed over a range of panel temperatures and insolation. In other work featuring simpler equivalent circuit

models R_s , was recalculated with temperature and insolation [14]. However, it has been illustrated in Section VI that the approach here of using fixed circuit parameters still provides a very good approximation of PV panel behaviour.

Most of these effects of temperature and insolation on photo current I_{ph} , dark saturation current I_0 and short circuit current I_{sc} can be considered linear, as can the effect of temperature on open circuit voltage, V_{oc} . These effects have been covered extensively in literature [9], [10], [11], [12], [13], [14], have been accounted for in the embedded Matlab model, Fig 10. However, V_{oc} varies logarithmically with insolation. As the graph in Fig 8 shows, the effect is observable and therefore must be accounted for, although this has been ignored in some other literature [14]. Therefore as this is a non-linear effect, its solution is dealt with iteratively and it becomes an extrinsic function (Fig 11) to the main embedded Matlab function describing the solar panel in section III.

III. EMBEDDED MATLAB FUNCTION FOR A SOLAR PANEL

The PV panel output voltage and current (as determined by its characteristic $V-I$ curve) will depend upon the panel's operating temperature and insolation. A given panel output voltage will give rise to a given output current, and vice versa and these will be determined at what point along the $V-I$ curve the panel is operating. From the perspective of modelling the PV panel, the choice is to model the output voltage as a function of output current (section IV), or the output current as a function of output voltage (section IV). The former of the two models is more suitable when modules are connected in series and share the same current. The latter model is better suited for the case when modules are connected in parallel and share the same voltage.

IV. 'VOLTAGE INPUT-CURRENT OUTPUT' MODEL FOR PV

The embedded Matlab Model is arranged as in Fig 3, where the panel output voltage is an input to the model and the outputs to the model are output current and power. As stated in section III, this model is more suited for cases where modules are connected in parallel and share the same voltage. However, they should also be used where they directly interface with a Simulink model element which has input current as one of its model inputs (e.g. a DC-DC converter performing maximum power point tracking).

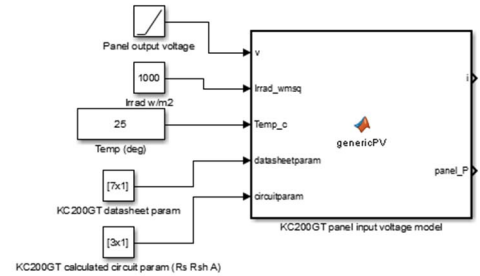


Fig 3. Generic panel 'voltage input, current output' model with datasheet and with calculated equivalent circuit parameters for KC200GT panel as inputs

The embedded Matlab code for the model in Fig 3 has been set out in Fig 10 and is based on (11) and (12). It must solve the rearranged expression for panel current (11) iteratively which is originally based on (1).

$$f(i) = 0 = I_{ph} - I_0 \left(e^{\frac{v+iR_s}{n_s V_t}} - 1 \right) - \frac{v+iR_s}{R_{sh}} - i \quad (11)$$

The Newton-Raphson iterative expression also requires the derivative of (11) with respect to panel current, i . This is expressed in (12).

$$f'(i) = -\frac{1}{n_s V_t} I_0 R_s \left(e^{\frac{v+iR_s}{n_s V_t}} - 1 \right) - \frac{R_s}{R_{sh}} - 1 \quad (12)$$

As an embedded Matlab function is not capable of handling the anonymous functions required for iterative solutions, the Newton-Raphson functions are written as extrinsic functions in a separate “.m” file which is declared in the embedded function and then called from it. The extrinsic function returns a value to the embedded Matlab code. The Matlab anonymous functions are illustrated in Fig 11 and Fig 12. Fig 4 provides an overview of this process between extrinsic files and the embedded Matlab.

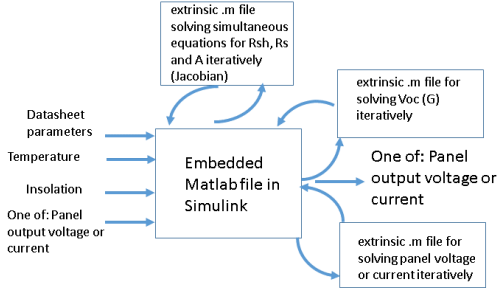


Fig 4. Overview of embedded Matlab model in Simulink with extrinsic function calls to Matlab .m files in the same path as the Simulink model

V. ‘CURRENT INPUT-VOLTAGE OUTPUT’ MODEL FOR PV

Using a similar method to the ‘voltage input-current output’ model in section IV, the solar panel can be modelled as an embedded Matlab model, where the panel output current represents the model input, and the voltage the model output as shown in Fig 5.

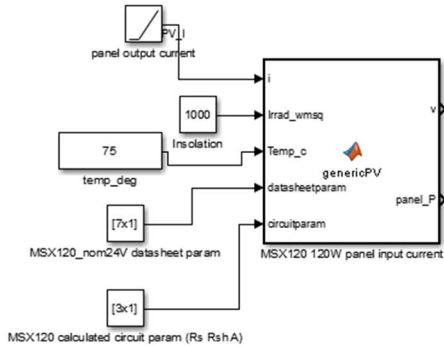


Fig 5. Generic panel ‘current input, voltage output’ model with calculated equivalent circuit parameters for MSX120 24V panel as inputs

The expression in (2) for panel voltage is rearranged and set to zero as in (13) so that it can be solved iteratively using Newton-Raphson, such panel output voltage for a given panel output current can be found.

$$f(v) = 0 = -v - R_{sh} \left[i - I_{ph} + I_0 \left(e^{\frac{v+iR_s}{n_s V_t}} - 1 \right) \right] - iR_s \quad (13)$$

The Newton-Raphson iterative expression also requires the derivative of (13) with respect to panel voltage, v , (14).

$$f'(v) = -\left[\frac{I_0 R_{sh}}{n_s V_t} \left(e^{\frac{v+iR_s}{n_s V_t}} \right) \right] - 1 \quad (14)$$

The Newton function is declared as an extrinsic ‘.m’ file (in separate Matlab file) as it contains anonymous functions which are not permitted in an embedded function within Matlab. An extrinsic Newton-Raphson Matlab function is called from within the Embedded Matlab function for this model in order to find the root of (13). An excerpt of this function is in Fig 6.

```
f=@(xx) -xx-Rsh*(I-Iph + I0*(exp((xx +
I*Rs)/(Ns*Vt_stc))-1))-I*Rs;
fd=@(xx) -(I0*Rsh*exp((xx +
I*Rs)/(Ns*Vt_stc)))/(Ns*Vt_stc)-1;
```

Fig 6. Code from extrinsic Matlab function solving for panel voltage, v which implements equations (13) and (14)

VI. PV PANEL MODEL VALIDATION

To test the accuracy of the modelling method, PV panel validation is illustrated here on three of the solar panel models whose equivalent circuit parameters were previously calculated in section II.C. Engage Digitizer software was used to extract data points from the datasheets to compare with the simulated V - I output characteristics of the relevant model. The accuracy of the data from the datasheet graphs for comparison is, however, limited to what can be achieved by extracting that data visually using Engage Digitizer.

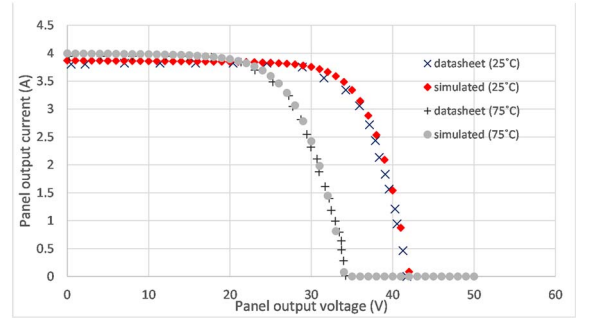


Fig 7. MSX120 24V nominal solar panel datasheet and simulated V - I characteristic at 25/75°C with 1000W/m² insolation [7]

The derived model for the KC200GT is compared with the KC200GT solar panel datasheet curves at different irradiation levels. The agreement between these datasheet and simulated figures is shown in Fig 8.

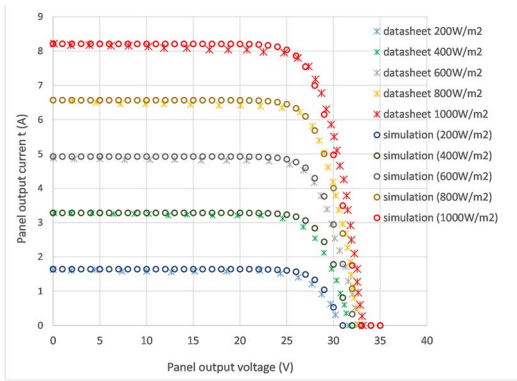


Fig 8. KC200GT solar panel datasheet graph figures and simulated V - I characteristic at 25°C over a range of insulations

Similarly the PV model SLP020 has been verified at different temperatures at standard insolation, based on the V - I output graph available on the SLP020 datasheet, as shown by Fig 9.

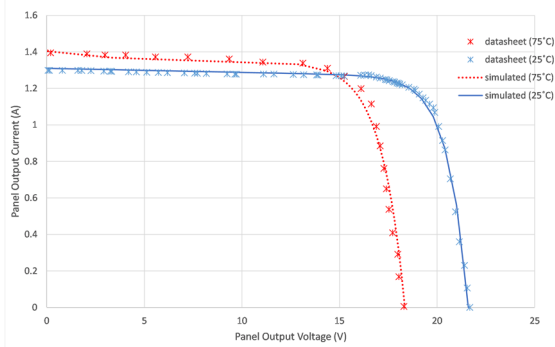


Fig 9. SLP020 12V nominal solar panel datasheet and simulated V - I characteristic at 25 and 75°C with 1000W/m² insolation [5]

VII. CONCLUSIONS

This paper has presented a simple means of producing a flexible embedded Matlab model for use within Simulink to represent a solar panel. Initially it uses simple iterative means of solving non-linear simultaneous equations to find equivalent circuit parameters at standard conditions, which are subsequently used as inputs into the PV model. It has been shown that the embedded Matlab model must call extrinsic functions to solve non-linear equations where anonymous functions are not permitted in embedded Matlab models. Panel current and voltage are interchangeable as inputs and outputs to the model such that it can provide flexibility in modelling with other Simulink elements. This research forms part of ongoing modelling work on hybrid DC and AC network micro grids powered from a range of locally available

sources including solar energy. Future work on this project will involve use of these models as part of DC microgrid design, coupling them with converters using maximum power point tracking and power sharing through DC droop control.

ACKNOWLEDGEMENTS

The authors are grateful for the funding of this research provided by a UK EPSRC Institutional Sponsorship Grant.

REFERENCES

- [1] The Economist, "Follow the sun", Published 16th April 2016 Accessed online: <http://www.economist.com/news/business/21696941-solar-power-reshaping-energy-production-developing-world-follow-sun>
- [2] N. J. Lewis, "Research opportunities to advance solar energy utilization", Science, Vol. 351, January 2016.
- [3] N. McNeill et al, "Efficient single-phase grid-tie inverter for small domestic photovoltaic scheme", in Proc. Power Electronics Machines and Drives Conference, 2012, Bristol.
- [4] Z. Yao et al, "Improved control strategy for the three-phase grid-connected inverter", IET Renewable Power Generation, Vol. 9, 2015, pp. 587-592.
- [5] SLP020 <http://www.mrsolar.com/content/pdf/SolarLand/Panels/SLP020-12U.pdf>
- [6] KC200GT datasheet <http://www.kyocera.com.sg/products/solar/pdf/kc200gt.pdf>
- [7] MSX120 datasheet accessed online at: http://www.soltec-solar.com/html/cms/bp/product_msx_120.pdf
- [8] Ci-Siang Tu Huan-Liang Tsai and Y.-J. Su, "Development of Generalized Photovoltaic Model Using MATLAB/SIMULINK," WCECS 2008, San Francisco, USA, 2008
- [9] G.Walker "Evaluating MPPT Converter Topologies Using a Matlab PV Model" Journal of Electrical & Electronics Engineering, Australia, Vol. 21, No. 1, 2001: 49-55
- [10] H. Bellia, R. Youcef, and M. Fatimab, "A detailed modeling of photovoltaic module using MATLAB," NRIAG Journal of Astronomy and Geophysics Volume 3, Issue 1, June 2014, Pages 53–61, 2014
- [11] D. Sera, R. Teodorescu, and P. Rodriguez, "PV panel model based on datasheet values," in 2007 IEEE International Symposium on Industrial Electronics, 2007, pp. 2392–2396
- [12] J. Y. Park and S. J. Choi, "Datasheet-based circuit parameter extraction method for maximum power point simulation of photovoltaic array," in 2015 9th ICPE-ECCE Asia, 2015, pp. 1413–1419.
- [13] J. Cubas, S. Pindado, and C. de Manuel, "Explicit Expressions for Solar Panel Equivalent Circuit Parameters Based on Analytical Formulation and the Lambert W-Function," Energies, vol. 7, no. 7, pp. 4098–4115, 2014
- [14] W. Xiao, W. G. Dunford, and A. Capel, "A novel modeling method for photovoltaic cells," in 2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No.04CH37551), 2004, vol. 3, pp. 1950–1956 Vol.3
- [15] G. E. Urroz, 2004 "Non linear Equations in Matlab" http://ocw.usu.edu/Civil_and_Environmental_Engineering/Numerical_Methods_in_Civil_Engineering/NonLinearEquationsMatlab.pdf,

APPENDIX

```
function [i,panel_P] = genericPV(v,Irrad_wmsq,Temp_c,datasheetparam,circuitparam)
%PREVIOUSLY CALCULATED CIRCUIT PARAMETERS ASSIGNED TO VARIABLES HERE Rs=circuitparam (etc.)
%CALCULATED PARAMATERS ASSIGN TO INTERNAL VARIABLES HERE - REMOVED FOR BREVITY]
%DATASHEET PARAMETERS = Isc, Voc, Impp, Vmpp, Ns, ki, kv. E.g. Isc=datasheetparam(1)
%DATASHEET PARAMETERS ARE ASSIGNED TO INTERNAL VARIABLES HERE - REMOVED FOR BREVITY]
G=Irrad_wmsq/1000;
k = 1.38e-23; % Boltzman's const
q = 1.60e-19; % charge on an electron
T_stc = 273 + 25; %temperature in Kelvin at standard operating conditions
T=Temp_c+273; %T is in Kelvin, not centigrade
Vt_stc=A*k*T_stc/q; %thermal voltage at standard conditions (STC)
```



```

Vt_T=(Vt_stc*T)/T_stc; %thermal voltage is scaled for temperature
%1. DARK SATURATION CURRENT Io AT STANDARD CONDITIONS
Io=(Isc-((Voc-Isc*Rs)/Rsh))*(exp(-Voc/(Ns*Vt_stc)));
%2. PHOTO CURRENT Iph AT STANDARD CONDITIONS
Iph=Io*exp(Voc/(Ns*Vt_stc))+Voc/Rsh;
Iph_G=Iph*G;
%3. SOLVING PANEL CURRENT BASED ON Iph and Io
%DEPENDENCE OF PHOTO CURRENT ON INSOLATION AND TEMPERATURE
Isc_G=Isc*G;
Isc_GT=Isc_G*(1+(ki/100)*(T-T_stc));
%%TEMPERATURE SCALING OF OPEN CIRCUIT VOLTAGE, Voc
Voc_T=Voc+kv*(T-T_stc);
%% IRRADIATION SCALING OF OPEN CIRCUIT VOLTAGE, Voc, SOLVE THIS WITH EXTRINSIC NEWTON RAPHSON SOLVER
coder.extrinsic('extrinsic_Voc');
Voc_param=[Iph_G,Io,Rsh,Ns,Vt_T];Voc_G=0;Voc_G=extrinsic_Voc(Voc,Voc_param);Voc_GT=Voc_G+kv*(T-T_stc);
%DEPENDENCE OF DARK SATURATION CURRENT ON INSOLATION AND TEMPERATURE
Io_GT=(Isc_GT-((Voc_GT-Isc_GT*Rs)/Rsh))*exp(-Voc_GT/(Ns*Vt_T));
Iph_GT=Io_GT*exp(Voc_GT/(Ns*Vt_T))+Voc_GT/Rsh;
%CALCULATE PANEL OUTPUT CURRENT AS A FUNCTION OF INPUT VOLTAGE
%LIMIT THE INPUT VOLTAGE TO OPEN CIRCUIT VOLTAGE
if (v > Voc_GT)v=Voc_GT; end;
% initial guess of output current - set equal to short circuit current
i_init=Isc_GT;
%DECLARE NEWTON FUNCTION AS EXTRINSIC .m FILE (file extrinsic to simulink file in same file path)
coder.extrinsic('extrinsic_newton');
newtonparam=[Rs,Rsh,Io,v,Ns,Vt_stc,Iph];
newtonparam=[Rs,Rsh,Io_GT,v,Ns,Vt_T,Iph_GT];
i=0; i=extrinsic_newton(i_init,newtonparam); panel_P=i*v;

```

Fig 10. Embedded Matlab within Simulink which accepts panel voltage as an input and outputs a panel current.

```

function [x]=extrinsic_Voc(x0,Vocparam)
% extrinsic newton-raphson algorithm to solve for Voc_G (non-linear)
Iph_G=Vocparam(1);Io=Vocparam(2);Rsh=Vocparam(3);Ns=Vocparam(4);Vt_T=Vocparam(5);
%%NEWTON RAPHSON SOLVING FOR 'xx' I.E. OPEN CIRCUIT VOLTAGE SCALED WITH IRRADIATION (Voc_G)
f=@(xx) log((Iph_G*Rsh-xx)/(Io*Rsh))*Ns*Vt_T-xx; fd=@(xx) (Ns*Vt_T)/(xx - Iph_G*Rsh) - 1;
%%NEWTON RAPHSON SOLVING FOR 'xx' I.E. Voc_G
N = 100; eps = 1.e-5; % define max. no. iterations and error
maxval = 10000.0; % define value for divergence
xx = x0;
while (N>0) xn = xx-f(xx)/fd(xx);
if abs(f(xn))<eps
    x=xn;
    return;
end; N = N - 1; xx = xn; end;

```

Fig 11. Extrinsic Matlab .m file which solves for open circuit voltage (Voc) as it varies with insolation, called from within embedded function in Fig 10

```

function [x]=extrinsic_newton(x0,newtonparam)
%'newtonparam' input parameters =[Rs,Rsh,Io_GT,v,Ns,Vt_T,Iph_GT]
[INTERNAL VARIABLES ASSIGNED TO VARIABLES HERE]
%%NEWTON RAPHSON SOLVING FOR 'xx' I.E. OUTPUT PANEL CURRENT BASED ON PANEL VOLTAGE AS INPUT
f=@(xx) Iph - xx - ((v + Rs*xx)/Rsh) - Io*(exp((v + Rs*xx)/(Ns*Vt_stc)) - 1);
fd=@(xx) - Rs/Rsh - (1/(Ns*Vt_stc))*Io*Rs*exp((v + Rs*xx)/(Ns*Vt_stc))-1;
[NEWTON RAPHSON SOLVER HERE- SAME AS FOR FIGURE 11]

```

Fig 12. Extrinsic Matlab .m file which solves for panel output current; called from within embedded function in Fig 10

$$f_1(R_{sh}, R_s, V_t) = I_{sc} - \frac{V_{mpp} + I_{mpp}R_s - I_{sc}R_s}{R_{sh}} - \left(I_{sc} - \frac{V_{oc} - I_{sc}R_s}{R_{sh}} \right) e^{\frac{V_{mpp} + I_{mpp}R_s - V_{oc}}{n_s V_t}} - I_{mpp} = 0 \quad (15)$$

$$f_2(R_{sh}, R_s, V_t) = I_{mpp} + V_{mpp} - \frac{\frac{-(I_{sc}R_{sh} - V_{oc} + I_{sc}R_s)e^{\frac{V_{mpp} + I_{mpp}R_s - V_{oc}}{n_s V_t}}}{n_s V_t R_{sh}} - \frac{1}{R_{sh}}}{1 + \frac{(I_{sc}R_{sh} - V_{oc} + I_{sc}R_s)e^{\frac{V_{mpp} + I_{mpp}R_s - V_{oc}}{n_s V_t}}}{n_s V_t R_{sh}} + \frac{R_s}{R_{sh}}} = 0 \quad (16)$$

$$f_3(R_{sh}, R_s, V_t) = \frac{\frac{-(I_{sc}R_{sh} - V_{oc} + I_{sc}R_s)e^{\frac{I_{sc}R_s - V_{oc}}{n_s V_t}}}{n_s V_t R_{sh}} - \frac{1}{R_{sh}}}{1 + \frac{(I_{sc}R_{sh} - V_{oc} + I_{sc}R_s)e^{\frac{I_{sc}R_s - V_{oc}}{n_s V_t}}}{n_s V_t R_{sh}} + \frac{R_s}{R_{sh}}} + \frac{1}{R_{sh}} = 0 \quad (17)$$